Post-Quantum TLS

Ostap Cherkashin

February 24, 2026

Agenda

1. Problem Overview
2. Impact on TLS
3. Post-Quantum Cryptography
4. Current Situation and Readiness

Part 1: Problem Overview

1.1 Security Notions

- Information-theoretic security: one-time pad, secret sharing

- Computational security: RSA, ECDH, AES, . . . , ML-KEM
  - NIST security levels correspond to AES-128, AES-192, AES-256
  - Level I is $\approx 2^{128}$ classical operations
  - Level III is $\approx 2^{192}$
  - Level V is $\approx 2^{256}$

1.2 Traditional Computational Problems

- Integer Factorization Problem: RSA encryption and signatures

$$n = p_1^{d_1} p_2^{d_2} \cdots p_t^{d_t}$$

- Discrete Logarithm Problem (DLP): Diffie-Hellman, ECDH, DSA

$$a = g^d \pmod{p}$$

1.3 Quantum Computers

- Optimism[i] and skepticism[ii] ...
- Small quantum computers are available on Azure[iii] and AWS[iv]
- In 2001 researches factored $15 = 3 \times 5$ on a quantum computer[v]
- Qubit, noise, decoherence $\longrightarrow$ "logical qubit"

---

[i] https://www.ionq.com/roadmap

[ii] Quantum computers: what are they good for? Springer Nature, 2023. https://www.nature.com/articles/d41586-023-01692-9

[iii] https://learn.microsoft.com/en-us/azure/quantum/qc-target-list

[iv] https://aws.amazon.com/braket/

[v] Vandersypen, L., Steffen, M., Breyta, G. et al. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. Nature 414, 883–887 (2001). https://doi.org/10.1038/414883a

1.4 Quantum Algorithms

- Shor's algorithm[i] solves discrete logarithm and integer factorization problems in polynomial time
  - Estimated number of logical qubits for ECDLP[ii] is about $8n$. This is about 2000 qubits for 256-bit curves.
  - Estimated number of logical qubits for factoring an n-bit integer[iii] is about $3n$. This is about 6000 qubits for RSA 2048.
- Grover's algorithm[iv] performs search over n records in $O(\sqrt{n})$

---

[i] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. Proceedings 35th Annual Symposium on Foundations of Computer Science. IEEE Comput. Soc. Press: 124–134. 1994.

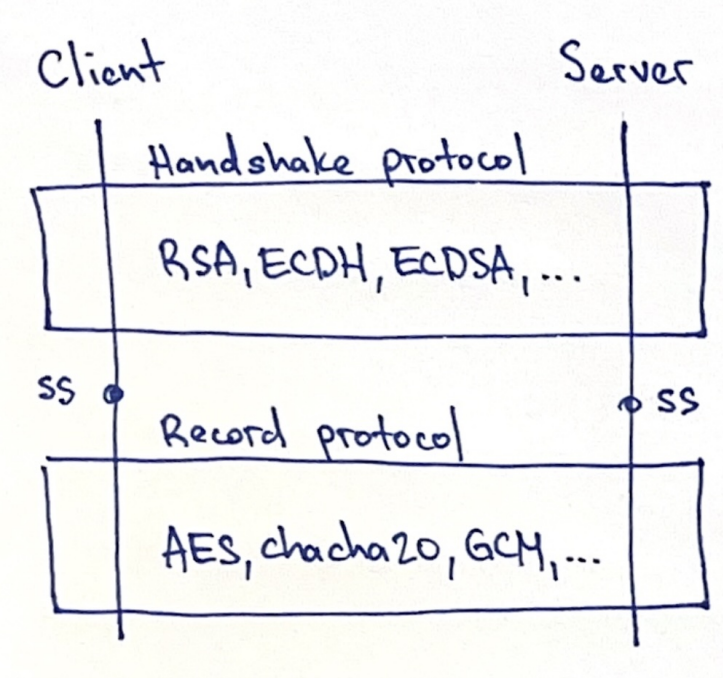
[ii] T. Häner, S. Jaques, M. Naehrig, M. Roetteler, M. Soeken, Improved Quantum Circuits for Elliptic Curve Discrete Logarithms, PQCrypto 2020, https://eprint.iacr.org/2020/077

[iii] C. Gidney and M. Ekera. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. Quantum, 5:433, April 2021. http://dx.doi.org/10.22331/q-2021-04-15-433

[iv] L.K. Grover. A fast quantum mechanical algorithm for database search. Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. STOC '96. Philadelphia, Pennsylvania, USA: Association for Computing Machinery: 212–219. 1996.

Part 2: Impact on TLS

2.1 TLS Refresher



- Handshake protocol produces a shared secret (public key cryptography)
- Shared secret is passed to the TLS key schedule algorithm (HKDF)
- Record protocol uses the shared secret (symmetric cryptography)
- TLS assures confidentiality and authenticity for data payload

2.2 Threats to Authentication in TLS

- Authentication assures that messages are not altered and come from the expected party
  - Digital signatures
  - Message authentication codes

- There are two modes of use in TLS:
  - “Static signatures” on certificate chains (RSA, ECDSA)
  - “Dynamic signatures” on handshake transcripts (RSA, ECDSA)

- Quantum threat: forging RSA and ECDSA signatures

- Impact: certificate chain compromise, man-in-the-middle (MITM)

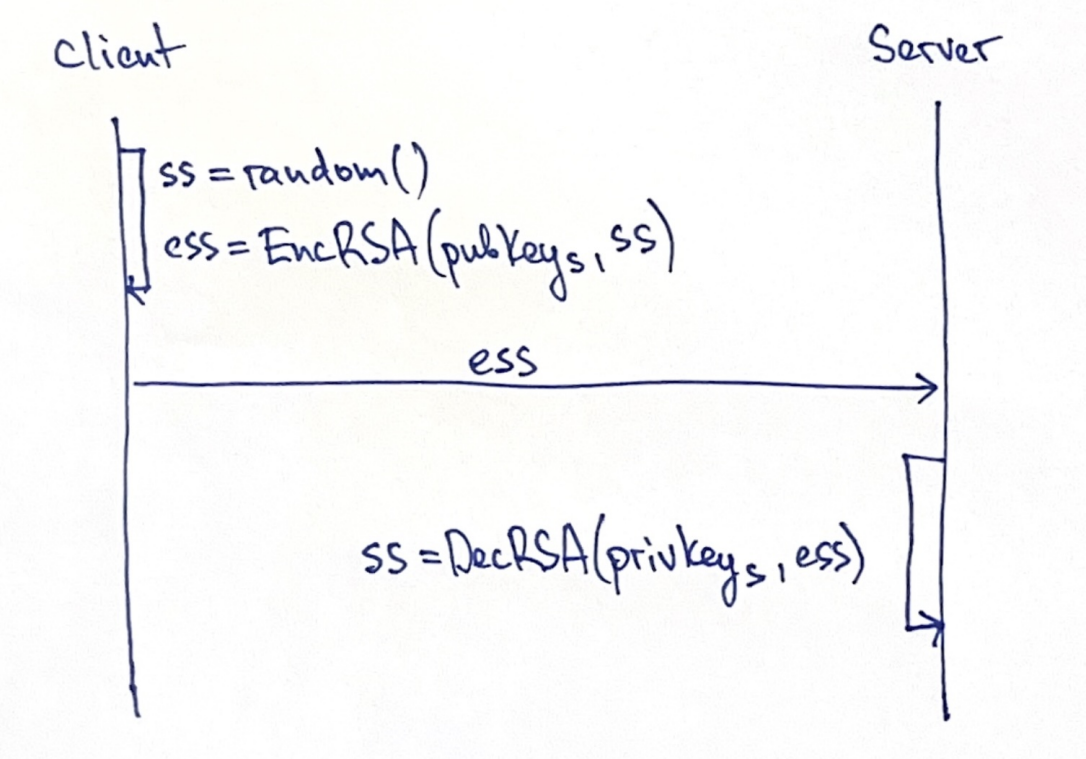
2.3 Threats to Confidentiality in TLS

- Confidentiality assures that only authorized parties have access to the information
  - Key exchange (Diffie-Hellman)
  - Symmetric encryption (AES)
  - Public key encryption (RSA-OAEP)

- Quantum threats:
  - Determining the shared secret by solving DLP
  - Optimizing search for the shared secret using Grover's algorithm

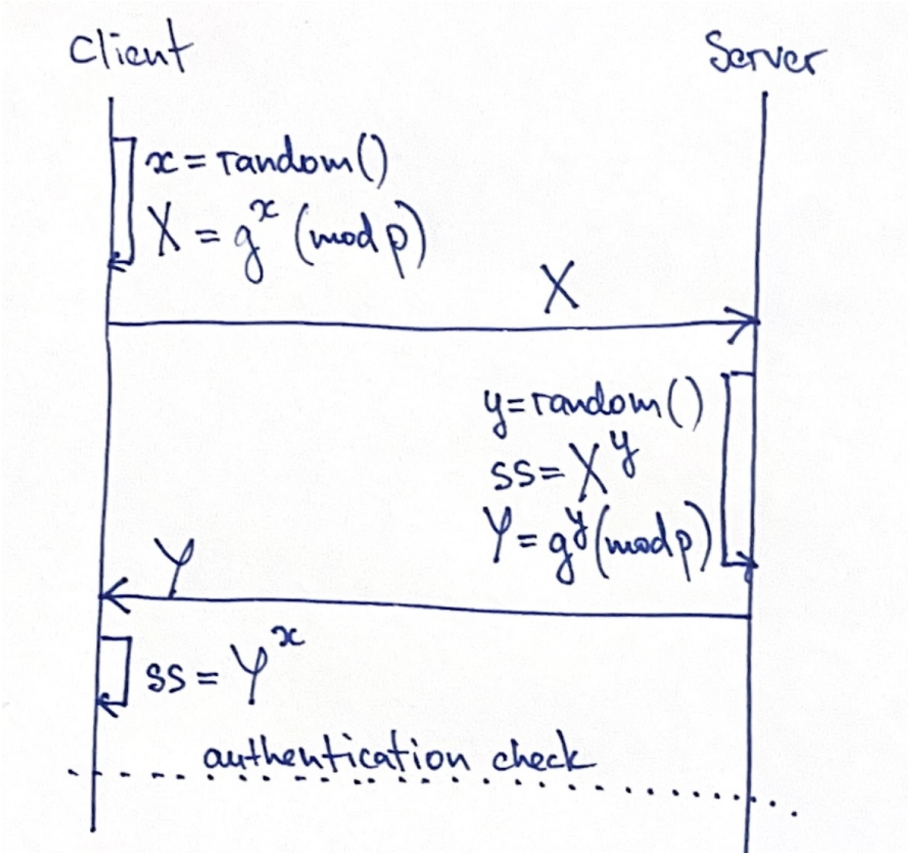
- Impact: confidentiality compromise for TLS payloads

2.4 RSA Key Exchange



- Available in TLSv1.2, but removed in TLSv1.3
- Not forward secure
- Quantum threat: attacker can decrypt shared secrets (ss)

2.5 Diffie-Hellman Key Exchange



- Shared secrets match: $X^y \equiv (g^x)^y \equiv (g^y)^x \equiv Y^x \pmod{p}$
- Inherently susceptible to MITM (requires authentication)
- Quantum threat: attacker can compute x from X and y from Y (DLP)

Part 3: Post-Quantum Cryptography

3.1 Classical Hardness Assumptions

- Integer factorization problem: given $n = p_1^{d_1} p_2^{d_2} \cdots p_t^{d_t}$, find $p_i^{d_i}$
  - Shor's algorithm efficiently solves this on a quantum computer

- Discrete logarithm problem: given g, p, and $a = g^d \pmod{p}$, find d
  - Shor's algorithm also solves this

- Need new hardness assumptions that are resilient to quantum computers

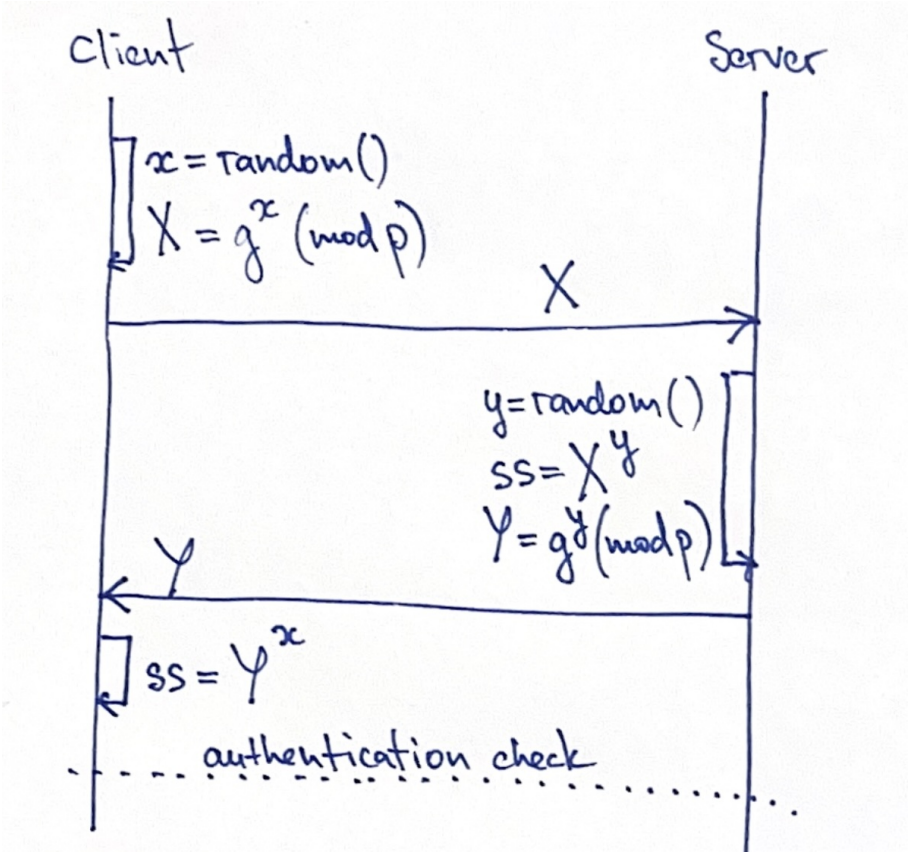
3.2 Post-Quantum Constructions[i] [ii]

Type	Signatures	Key Agreements	Hardness Assumptions
Codes	LESS (c)	Classic McEllice (c), BIKE (c)	SDP, PEP
Isogenies	SQISign (c)	~~SIKE~~ (b), CSIDH	IP
Lattices	ML-DSA (s)	ML-KEM (s)	SVP
Multivariate	UOV (c), MAYO (c)		MQ

[i] SDP - Syndrome Decoding Problem, PEP - Permutation Equivalence Problem, IP - Isogeny Problem, SVP - Shortest Vector Problem, MQ - Multivariate Quadratic Problem

[ii] (s) - standardized, (c) - candidate for standardization, (b) - broken

3.3 Post-Quantum Diffie-Hellman?



- Idea: swap the discrete logarithm problem (DLP) for something harder
- Cryptographic group action $\psi : S \times G \to S$
- Easy to compute $\psi(g, x)$, but hard to find x given $\psi(g, x)$
- Shared secret $\psi(\psi(g, x), y) = \psi(g, x \cdot y) = \psi(g, y \cdot x) = \psi(\psi(g, y), x)$

3.4 Diffie-Hellman Based on Isogenies?

- Isogeny is a map between two elliptic curves $\varphi : E \to E'$
  - Given a point P on E_1 it is easy to compute[i] $\varphi_P : E_1 \to E_P$
  - Given E_1 and E_P it is believed to be hard to compute φ_P

- This yields a cryptographic group action $\varphi : S \times G \to S$
  - S is a set of elliptic curves
  - G is an ideal class group (think sets of points on arbitrary curves)

- Algorithms are still in research phase . . .
  - CSIDH is considered slow for secure parameter sets[ii]
  - Attack on SIDH[iii] turned into a powerful optimization tool[iv]
  - New algorithms are closing the performance gap[v] [vi]

3.5 References for ”Diffie-Hellman Based on Isogenies?”

i. J. Vélu. Isogénies entre courbes elliptiques. Comptes Rendus de l’Académie des Sciences de Paris, 273:238–241, 1971.

ii. X. Bonnetain and A. Schrottenloher. Quantum Security Analysis of CSIDH. EUROCRYPT 2020, https://eprint.iacr.org/2018/537

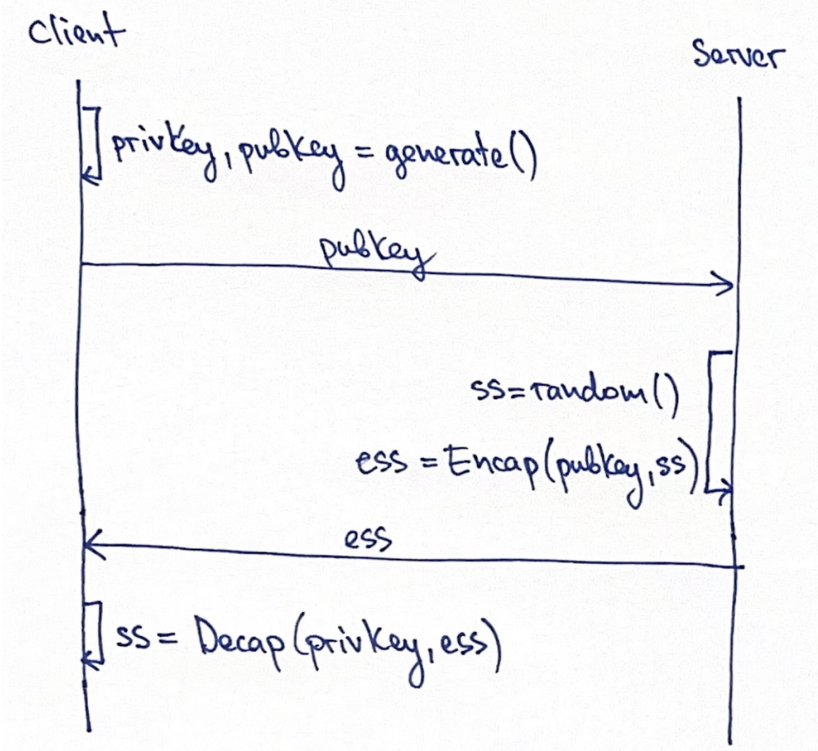
iii. W. Castryck, T. Decru. An efficient key recovery attack on SIDH. EUROCRYPT 2023. https://eprint.iacr.org/2022/975

iv. D. Robert. Evaluating isogenies in polylogarithmic time. Cryptology ePrint Archive. https://eprint.iacr.org/2022/1068

v. P. Dartois, J. Komada Eriksen, R. Invernizzi, F. Vercauteren. qt-Pegasis: Simpler and Faster Effective Class Group Actions. EUROCRYPT 2026. https://eprint.iacr.org/2025/1859

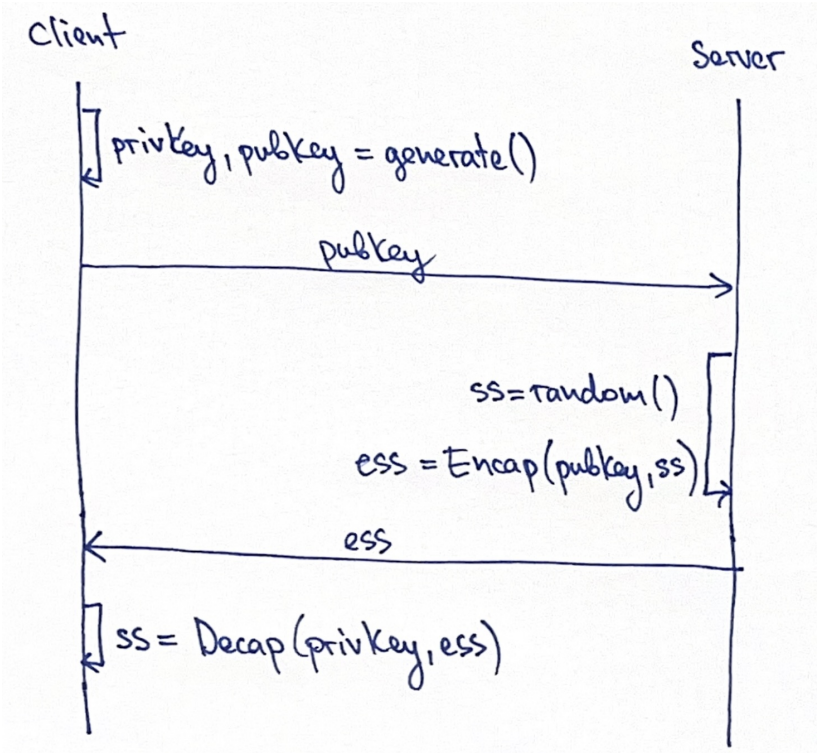
vi. M. Houben. Efficient post-quantum commutative group actions from orientations of large discriminant. ASIACRYPT 2025. https://eprint.iacr.org/2025/1098

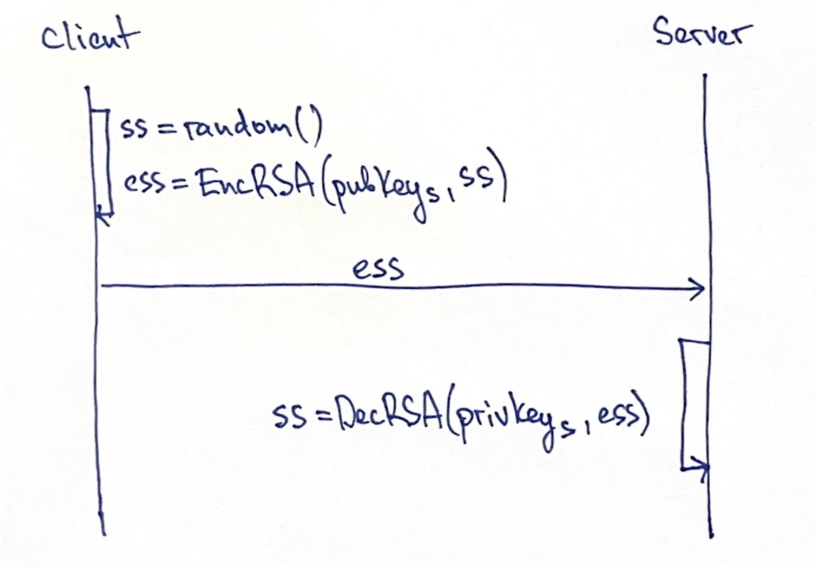
3.6 Key Encapsulation Mechanisms



- Forward secrecy is established via ephemeral privKey, pubKey
- Diffie-Hellman can also be described in terms of KEM
  - Client public key $X = g^x \pmod p$
  - Client private key x
  - Encapsulated shared secret Y

3.7 KEM vs RSA Key Exchange





- KEM generates a new key pair for every encapsulation
- Deprecated RSA key exchange reuses the server keys

3.8 Hybrid Key Encapsulation Mechanisms

1. Client generates two key pairs and sends both public keys to the server

$$\mathsf{pubKey}_1, \mathsf{privKey}_1 = \mathsf{ECDH.KeyGen}()$$
$$\mathsf{pubKey}_2, \mathsf{privKey}_2 = \mathsf{MLKEM.KeyGen}()$$

2. Server computes shared secrets and sends encapsulations back

$$\mathsf{sharedSecret}_1, \mathsf{encryptedSharedSecret}_1 = \mathsf{ECDH.Encaps}(\mathsf{pubKey}_1)$$
$$\mathsf{sharedSecret}_2, \mathsf{encryptedSharedSecret}_2 = \mathsf{MLKEM.Encaps}(\mathsf{pubKey}_2)$$

3. Client decapsulates the secrets

$$\mathsf{sharedSecret}_1 = \mathsf{ECDH.Decaps}(\mathsf{encryptedSharedSecret}_1, \mathsf{privKey}_1)$$
$$\mathsf{sharedSecret}_2 = \mathsf{MLKEM.Decaps}(\mathsf{encryptedSharedSecret}_2, \mathsf{privKey}_2)$$

4. Shared secrets are combined on both ends

$$\mathsf{sharedSecret} = \mathsf{TLS.KeySchedule}(\mathsf{sharedSecret}_1, \mathsf{sharedSecret}_2, \dots)$$

Part 4: Current Situation and Readiness

4.1 Where Are We?

- One can collect encrypted TLS traffic and wait for a quantum computer
- Then run Shor's algorithm to break ECDH and decrypt the payloads
- Post-Quantum Cryptography (PQC) is believed to be resilient to this
- Hybrid KEM safeguards the adoption of PQC

4.2 Post-Quantum TLS Handshake

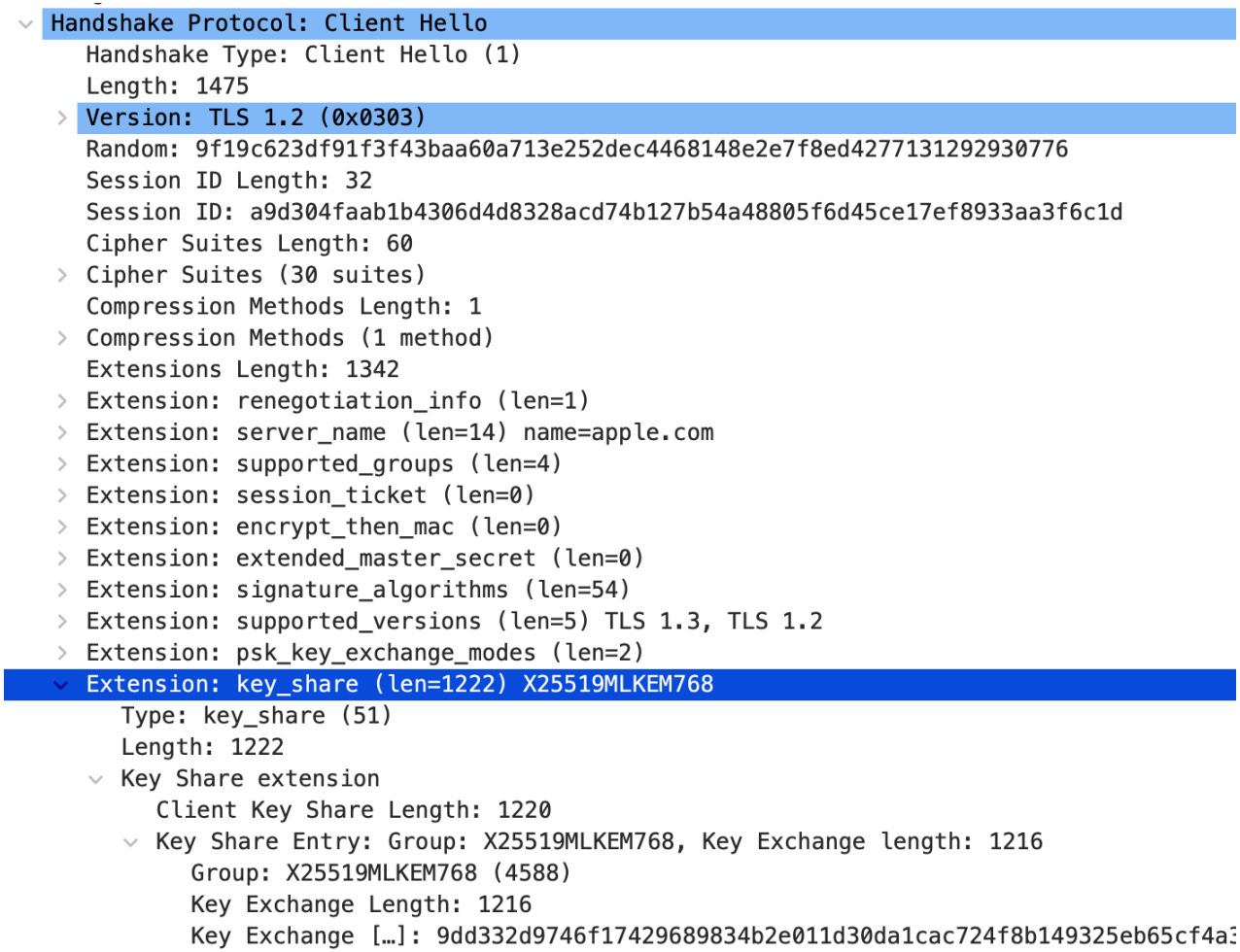
- OpenSSL supports hybrid KEM since 3.5.0 (April 8, 2025)

- Some TLS servers already support hybrid KEMs

```
$ openssl s_client \
        -connect apple.com:443 \
        -groups X25519MLKEM768
...
Negotiated TLS1.3 group: X25519MLKEM768
...
```

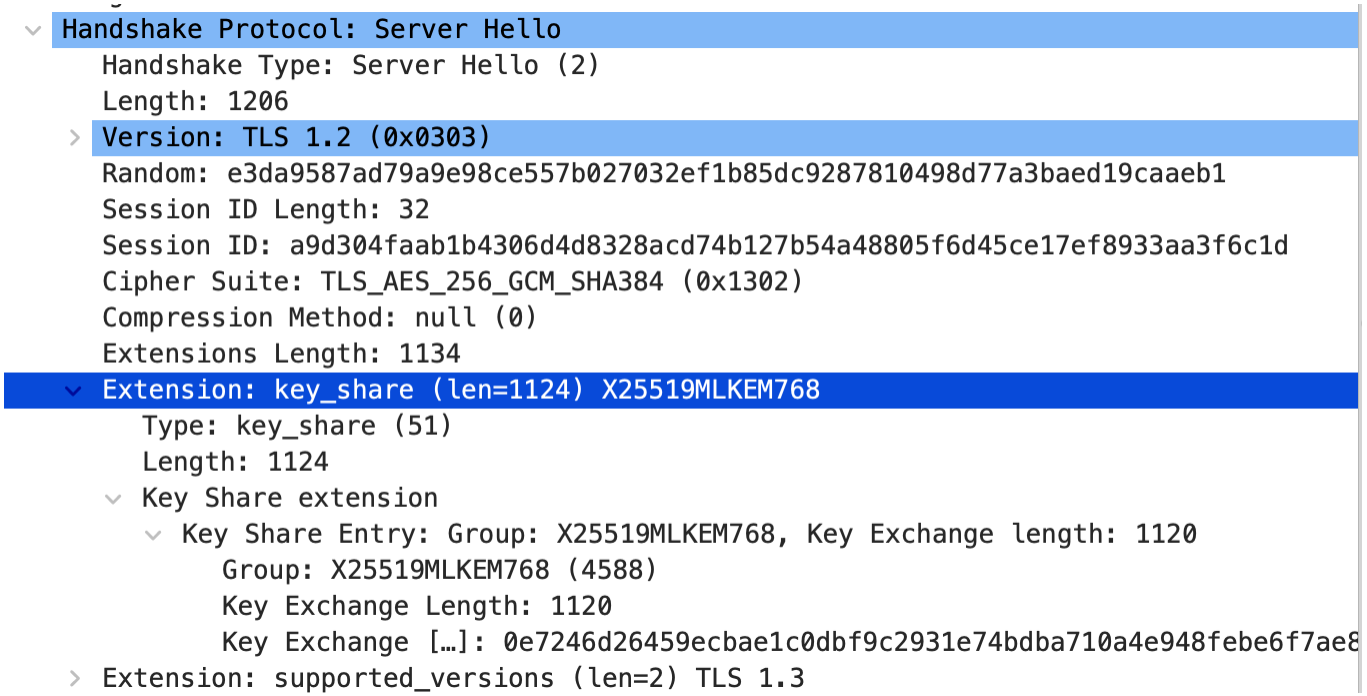
- IETF RFCs are submitted for publication
  - `https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/`
  - `https://datatracker.ietf.org/doc/draft-ietf-tls-ecdhe-mlkem/`

4.3 Post-Quantum ClientHello



- `supported_groups` signals support for hybrid KEM
- `key_share` carries both public keys (≈1KB larger)

4.4 Post-Quantum ServerHello



- `key_share` carries both encapsulated shared secrets
- The message is also ≈1KB larger

4.5 Performance

- Main characteristics: speed, key size, ciphertext size
- PQC requires more data transfer and more computation
- Speed is on par with traditional schemes[i]
- Overall impact on performance appears to be $\approx$10%[ii]

---

[i] N. Alnahawi, J. Müller, J. Oupicky, and A. Wiesmaier. SoK: Post-quantum TLS handshake. Cryptology ePrint Archive. https://eprint.iacr.org/2023/1873.

[ii] M. Anastasova and P. Kampanakis. MLWE's impact on Web Metrics, mTLS TTLB, and AWS service endpoint connections. Cryptology ePrint Archive. https://eprint.iacr.org/2025/2235

4.6 Standards

- NIST Post-Quantum Cryptography Work Group[i]
  - Standardization started in 2016
  - First set of finalists announced in 2022
  - FIPS 203, 204, 205 were published in August 2024
  - Round 4 evaluation is still ongoing (KEMs only)
  - Additional signature scheme evaluation started in 2023
- IETF Work Groups
  - TLS WG is working on further PQC standards[ii]
  - JOSE WG is bringing PQC signatures and encryption for JWTs[iii]
- W3C is considering how to handle XML and SAML[iv]

---

[i] `https://www.nist.gov/pqcrypto`

[ii] `https://datatracker.ietf.org/wg/tls/documents/`

[iii] `https://datatracker.ietf.org/wg/jose/documents/`

[iv] `https://github.com/w3c/strategy/issues/484`

4.7 Simple Things You Can Improve Today

- Tune symmetric cryptography against Grover's algorithm
  - AES-128 $\rightarrow$ AES-256
  - SHA-256 $\rightarrow$ SHA-512

- Increase asymmetric keys to require more qubits
  - 3072-bit RSA (6000 $\rightarrow$ 9000 qubit[i])
  - 500-bit elliptic curves (2000 $\rightarrow$ 4000 qubit)

[i] This is based on the estimates referenced in Part 1.

Thank You